

FFFFFFFFFFF	111	111	AAAAAAA
FFFFFFFFFFF	111	111	AAAAAAA
FFFFFFFFFFF	111	111	AAAAAAA
FFF	111111	111111	AAA
FFF	111111	111111	AAA
FFF	111111	111111	AAA
FFF	111	111	AAA
FFF	111	111	AAA
FFF	111	111	AAA
FFF	111	111	AAA
FFFFFFFFFFF	111	111	AAA
FFFFFFFFFFF	111	111	AAA
FFFFFFFFFFF	111	111	AAA
FFF	111	111	AAAAAA
FFF	111	111	AAAAAA
FFF	111	111	AAAAAA
FFF	111	111	AAA
FFF	111	111	AAA
FFF	11111111	11111111	AAA
FFF	11111111	11111111	AAA
FFF	11111111	11111111	AAA

FILEID**MAKACC

F 1

MM MM AAAAAAA KK KK AAAAAAA CCCCCCCC CCCCCCCC
MM MM AAAAAAA KK KK AA AA CC CC
MM MM AA AA KK KK AA AA CC CC
MM MM AA AA KK KK AA AA CC CC
MM MM AA AA KKKKKK AA AA CC CC
MM MM AA AA KKKKKK AA AA CC CC
MM MM AAAAAAAAAA KK KK AAAAAAAAAA CC CC
MM MM AAAAAAAAAA KK KK AAAAAAAAAA CC CC
MM MM AA AA KK KK AA AA CC CC
MM MM AA AA KK KK AA AA CC CC
MM MM AA AA KK KK AA AA CC CC
MM MM AA AA KK KK AA AA CC CC

LL IIIII SSSSSSSS
LL IIIII SSSSSSSS
LL II SS SS
LLLLLLLLL IIIII SSSSSSSS
LLLLLLLLL IIIII SSSSSSSS

MAK
VO4

```
0001 0 MODULE MAKACC (
0002 0           LANGUAGE (BLISS32),
0003 0           IDENT = 'V04-000',
0004 0           ) =
0005 1 BEGIN
0006 1
0007 1
0008 1 ****
0009 1 *
0010 1 * COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0011 1 * DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0012 1 * ALL RIGHTS RESERVED.
0013 1 *
0014 1 * THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0015 1 * ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0016 1 * INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0017 1 * COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0018 1 * OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0019 1 * TRANSFERRED.
0020 1 *
0021 1 * THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0022 1 * AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0023 1 * CORPORATION.
0024 1 *
0025 1 * DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0026 1 * SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0027 1 *
0028 1 *
0029 1 ****
0030 1
0031 1 ++
0032 1
0033 1 FACILITY: F11ACP Structure Level 1
0034 1
0035 1 ABSTRACT:
0036 1
0037 1 This routine makes the necessary hookups in the I/O database to
0038 1 reflect a new file access.
0039 1 ENVIRONMENT:
0040 1
0041 1 STARLET operating system, including privileged system services
0042 1 and internal exec routines. This routine must be called
0043 1 in kernel mode.
0044 1 --
0045 1
0046 1
0047 1
0048 1 AUTHOR: Andrew C. Goldstein, CREATION DATE: 20-Dec-1976 17:28
0049 1
0050 1 MODIFIED BY:
0051 1
0052 1 V02-001 LMP0005          L. Mark Pilant,        29-Dec-1981 15:05
0053 1                         Add support for Cathedral windows.
0054 1
0055 1 A0100   ACG0001          Andrew C. Goldstein,    10-Oct-1978 20:01
0056 1                         Previous revision history moved to F11A.REV
0057 1
```

MAKACC
VO4-000

H 1
16-Sep-1984 01:09:33 14-Sep-1984 12:29:42 VAX-11 BLiss-32 V4.0-742
DISK\$VMSMASTER:[F11A.SRC]MAKACC.B32;1 Page 1

58 0058 1 !**
59 0059 1
60 0060 1
61 0061 1 LIBRARY 'SYS\$LIBRARY:LIB.L32';
62 0062 1 REQUIRE 'SRC\$:FCPDEF.B32';

MAK
VO4

64 0377 1 GLOBAL ROUTINE MAKE_ACCESS (FCB, WINDOW, ABD) : NOVALUE =
65 0378 1 !++
66 0379 1 FUNCTIONAL DESCRIPTION:
67 0380 1 This routine makes the necessary hookups in the I/O database to
68 0381 1 reflect a new file access.
69 0382 1 CALLING SEQUENCE:
70 0383 1 MAKE_ACCESS (ARG1, ARG2, ARG3)
71 0384 1 INPUT PARAMETERS:
72 0385 1 ARG1: address of FCB to access
73 0386 1 ARG2: address of window to link up
74 0387 1 ARG3: address of buffer descriptors
75 0388 1 IMPLICIT INPUTS:
76 0389 1 CURRENT_VCB: VCB of volume in process
77 0390 1 OUTPUT PARAMETERS:
78 0391 1 NONE
79 0392 1 IMPLICIT OUTPUTS:
80 0393 1 NONE
81 0394 1 ROUTINE VALUE:
82 0395 1 NONE
83 0396 1 SIDE EFFECTS:
84 0397 1 VCB transaction count bumped, access counts in FCB adjusted,
85 0398 1 FCB and window hooked up.
86 0399 1 --
87 0400 1 BEGIN
88 0401 1 MAP
89 0402 1 FCB : REF BBLOCK, ! FCB arg
90 0403 1 WINDOW : REF BBLOCK, ! window arg
91 0404 1 ABD : REF BBLOCKVECTOR [,ABD\$C_LENGTH];
92 0405 1 ! buffer descriptor arg
93 0406 1 LOCAL
94 0407 1 WINDOW_SEGMENT : REF BBLOCK; ! address of the current window segment
95 0408 1 EXTERNAL
96 0409 1 CLEANUP_FLAGS : BITVECTOR, ! cleanup action flags
97 0410 1 CURRENT_VCB : REF BBLOCK, ! VCB
98 0411 1 PMSSGL_OPEN : ADDRESSING_MODE (ABSOLUTE),
99 0412 1 ! system count of currently open files
100 0413 1 PMSSGL_OPENS : ADDRESSING_MODE (ABSOLUTE);
101 0414 1 ! system count of files opened
102 0415 1 ! If the access count in the FCB is zero, hook it into the FCB list,
103 0416 1 since it is not there yet. If, however, the directory LRU
104 0417 1 bit is set, the FCB is already in the list. Then clear the bit and
105 0418 1
106 0419 1
107 0420 1
108 0421 1
109 0422 1
110 0423 1
111 0424 1
112 0425 1
113 0426 1
114 0427 1
115 0428 1
116 0429 1
117 0430 1
118 0431 1
119 0432 1
120 0433 1

```
121 0434 2 ! credit an entry to the LRU count.  
122 0435 2 !  
123 0436 2  
124 0437 2 IF .FCB[FCBSW_ACNT] EQL 0  
125 0438 2 THEN  
126 0439 3 BEGIN  
127 0440 3 IF NOT .FCB[FCBSV_DIR]  
128 0441 3 THEN INSQUE (.FCB, .CURRENT_VCB[VCB$L_FCBBL])  
129 0442 3 ELSE  
130 0443 4 BEGIN  
131 0444 4 FCB[FCBSV_DIR] = 0;  
132 0445 4 CURRENT_VCB[VCB$B_LRU_LIM] = .CURRENT_VCB[VCB$B_LRU_LIM] + 1;  
133 0446 3 END;  
134 0447 2 END;  
135 0448 2  
136 0449 2 ! Now hook the window onto the FCB and adjust the access counts  
137 0450 2 ! according to the access control bits in the window.  
138 0451 2 !  
139 0452 2  
140 0453 2 WINDOW_SEGMENT = .WINDOW;  
141 0454 2 DO  
142 0455 3 BEGIN  
143 0456 3 INSQUE (.WINDOW_SEGMENT, .FCB[FCBSL_WLBL]);  
144 0457 3 WINDOW_SEGMENT = .WINDOW_SEGMENT[WCBSL_LINK];  
145 0458 3 END  
146 0459 2 UNTIL .WINDOW_SEGMENT EQL 0;  
147 0460 2 FCB[FCBSW_ACNT] = .FCB[FCBSW_ACNT] + 1; ! bump access count  
148 0461 2  
149 0462 2 IF .WINDOW[WCBSV_NOREAD]  
150 0463 2 THEN FCB[FCBSV_EXCL] = 1; ! set exclusive access  
151 0464 2  
152 0465 2 IF .WINDOW[WCBSV_NOWRITE]  
153 0466 2 THEN FCB[FCBSW_LCNT] = .FCB[FCBSW_LCNT] + 1; ! no writers  
154 0467 2  
155 0468 2 IF .WINDOW[WCBSV_NOTRUNC]  
156 0469 2 THEN FCB[FCBSW_TCNT] = .FCB[FCBSW_TCNT] + 1; ! no truncates  
157 0470 2  
158 0471 2 ! For a write access, bump the writer count. If this is the  
159 0472 2 first write, and the file is the index file or the storage map, set  
160 0473 2 the appropriate flag in the VCB.  
161 0474 2 !  
162 0475 2  
163 0476 2 IF .WINDOW[WCBSV_WRITE]  
164 0477 2 THEN  
165 0478 3 BEGIN  
166 0479 3 IF .FCB[FCBSW_WCNT] EQL 0  
167 0480 3 THEN  
168 0481 4 BEGIN  
169 0482 4 IF .FCB[FCBSW_FID_NUM] EQL 1  
170 0483 4 THEN CURRENT_VCB[VCBSV_WRITE_IF] = 1;  
171 0484 4 IF .FCB[FCBSW_FID_NUM] EQL 2  
172 0485 4 THEN CURRENT_VCB[VCBSV_WRITE_SM] = 1;  
173 0486 3 END;  
174 0487 3 FCB[FCBSW_WCNT] = .FCB[FCBSW_WCNT] + 1;  
175 0488 2 END;  
176 0489 2  
177 0490 2 ! Count the access in the volume transaction count and return
```

```

178 0491 2 ! the window address for the user's CCB.
179 0492 2 !
180 0493 2
181 0494 2 PMSSGL_OPEN = .PMSSGL_OPEN + 1;           ! bump open file count
182 0495 2 PMSSGL_OPENS = .PMSSGL_OPENS + 1;        ! bump count of opens
183 0496 2 CURRENT_VCB[VCB$W_TRANS] = .CURRENT_VCB[VCB$W_TRANS] + 1;
184 0497 2
185 0498 2 ABD[ABD$C_WINDOW, ABD$W_COUNT] = 4;      ! enable write-back
186 0499 2 .ABD[ABD$C_WINDOW, ABD$W_TEXT] + ABD[ABD$C_WINDOW, ABD$W_TEXT] + 1 = .WINDOW;
187 0500 2 ! put window address in buffer text
188 0501 2
189 0502 2 ! Mark the access complete in the cleanup action flags.
190 0503 2 !
191 0504 2
192 0505 2 CLEANUP_FLAGS[CLF_DEACCESS] = 1;
193 0506 2
194 0507 1 END;                                     ! end of routine MAKE_ACCESS

```

```

.TITLE MAKACC
.IDENT \V04-000\
.EXTRN CLEANUP_FLAGS, CURRENT_VCB
.EXTRN PMSSGL_OPEN, PMSSGL_OPENS
.PSECT $CODE$, NOWRT, 2

```

				.ENTRY	MAKE_ACCESS, Save R2,R3	0377
	53	0000G	000C 00000	MOVAB	CURRENT_VCB, R3	
	51	04	CF 9E 00002	MOVL	FCB, R1	0437
		1A	AC D0 00007	TSTW	26(R1)	
			14 B5 0000B	BNEQ	2\$	
			12 0000E	MOVL	CURRENT_VCB, R0	0441
	50	63	D0 00010	BLBS	34(R1), -1\$	0440
	06	22	A1 E8 00013	INSQUE	(R1), @4(R0)	0441
	04	B0	61 0E 00017	BRB	2\$	
			07 11 0001B	BICB2	#1, 34(R1)	0444
	22	A1	01 8A 0001D	INC B	73(R0)	0445
			49 A0 96 00021	MOVL	WINDOW, WINDOW_SEGMENT	0453
		51	08 AC DO 00024	MOVL	FCB, R0	0456
		50	04 AC DO 00028	INSQUE	(WINDOW_SEGMENT), @20(R0)	
	14	B0	61 0E 0002C	MOVL	32(WINDOW_SEGMENT), WINDOW_SEGMENT	0457
		51	20 A1 DO 00030	BNEQ	3\$	0459
			F2 12 00034	MOVL	FCB, R0	0460
		50	04 AC DO 00036	INCW	26(R0)	
			1A A0 B6 0003A	MOVL	WINDOW, R2	0462
		52	08 AC DO 0003D	BBC	#2, 21(R2), 4\$	
	04	15	A2 02 E1 00041	BIS B2	#8, 34(R0)	0463
		22	A0 08 88 00046	BLBC	20(R2), 5\$	0465
		03	14 A2 E9 0004A	INCW	30(R0)	0466
			4\$: 1E A0 B6 0004E	BBC	#3, 21(R2), 6\$	0468
	03	15	A2 03 E1 00051	INCW	32(R0)	0469
			5\$: 20 A0 B6 00056	BBC	#1, 11(R2), 9\$	0476
	22	08	A2 01 E1 00059	TSTW	28(R0)	0479
			6\$: 1C A0 B5 0005E	BNEQ	8\$	
			1A 12 00061	CMPW	36(R0), #1	0482
		01	24 A0 B1 00063	BNEQ	7\$	
			07 12 00067			

MAK
Sym
AQB
BAD
BIT
DIG
DIR
DOT
FCB
FIB
FIB
FIB
HEA
HI
IND
LC
LEN
LIB
LO
MAR
MVL
NAM
NMB
NMB
NMB
NMB
NMB
NMB
NMB
NMB
NMB
PER
RVT
SEM
SSS
SSS
STA
STR
TYP
UC
VCB
WCB
PSE
---\$AE
\$CC

0B	51	63	D0	00069	MOVL	CURRENT_VCB, R1	: 0483	
	A1	01	88	0006C	BISB2	#1, 11(R1)		
02	24	A0	B1	00070	7\$: CMPW	36(R0), #2	: 0484	
		07	12	00074	BNEQ	8\$		
0B	51	63	D0	00076	MOVL	CURRENT_VCB, R1	: 0485	
	A1	02	88	00079	BISB2	#2, 11(R1)		
		1C	A0	B6	0007D	8\$: INCW	: 0487	
		00000000G	9F	D6	00080	9\$: INCL	: 0494	
		00000000G	9F	D6	00086	INCL	: 0495	
	50	63	D0	0008C	MOVL	CURRENT_VCB, R0	: 0496	
		OC	A0	B6	0008F	INCW	: 0497	
02	51	AC	D0	00092	MOVL	ABD, R1	: 0498	
	A1	04	B0	00096	MOVW	#4, 2(R1)		
	50	61	3C	0009A	MOVZWL	(R1), R0	: 0499	
		01	A140	9F	0009D	PUSHAB	1(R1)[R0]	
	9E	52	D0	000A1	MOVL	R2, @(SP)+		
	0000G	CF	01	88	000A4	BISB2	#1, CLEANUP_FLAGS+2	: 0505
				04	000A9	RET	: 0507	

: Routine Size: 170 bytes, Routine Base: \$CODE\$ + 0000

```
: 195      0508 1
: 196      0509 1 END
: 197      0510 0 ELUDOM
```

PSECT SUMMARY

Name	Bytes	Attributes
\$CODE\$	170	NOVEC,NOWRT, RD , EXE,NOSHR, LCL, REL, CON,NOPIC,ALIGN(2)

Library Statistics

File	----- Symbols -----	Pages Mapped	Processing Time
	Total Loaded Percent		
\$_\$255\$DUA28:[SYSLIB]LIB.L32;1	18619 23 0	1000	00:02.0

COMMAND QUALIFIERS

BI ISS/CHECK=(FIELD,INITIAL,OPTIMIZE)/LIS=LIS\$:\$MAKACC/OBJ=OBJ\$:\$MAKACC MSRC\$:\$MAKACC/UPDATE=(ENH\$:\$MAKACC)

MAKACC
V04-000

M 1
16-Sep-1984 01:09:33 VAX-11 Bliss-32 V4.0-742

Page 7

: Size: 170 code + 0 data bytes
: Run Time: 00:08.2
: Elapsed Time: 00:26.4
: Lines/CPU Min: 3754
: Lexemes/CPU-Min: 16181
: Memory Used: 114 pages
: Compilation Complete

MAP
VA)

Pha

Ini
Con
Pas
Syn
Pas
Syn
Pse
Crc
Ass

The
183
The
489
13

Mac

-\$2
-\$2
T01
278
The
MAC

0166 AH-BT13A-SE
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION
CONFIDENTIAL AND PROPRIETARY

MODIFY
LIS

REQUE
LIS

RWATTR
LIS

SCHFCB
LIS

MAKREC
LIS

MPWIND
LIS

MAPUBN
LIS

PMS
LIS

ROHEDR
LIS

RWUB
LIS

SMALOC
LIS

ROBLOK
LIS

RETDIR
LIS

MOUNT
LIS

NXTHOR
LIS

MARKHIB
LIS

MAKSTR
LIS